

Quasi-Static Legged Locomotors as Nonholonomic Systems

Joel Burdick
Mechanical Engineering, Caltech
jwb@robotics.caltech.edu

Bill Goodwine
Aerospace/Mechanical Engineering, Notre Dame
jgoodwin@nd.edu

Abstract: We show how motion planning and control ideas for smooth nonholonomic systems can be extended to legged quasi-static locomotion via the notion of “stratified” configuration spaces and “stratified” control theory. We particularly consider “minimalist” legged systems, which are not well handled by conventional theories based on foot placement. We briefly discuss controllability issues, and then present a motion planning algorithm for stratified systems. The method does not depend upon the number of legs, nor is it based on foot placement concepts.

1 Introduction

This paper’s first goal is to show how motion planning and control concepts that have been developed for wheeled nonholonomic vehicles can be formally extended to quasi-static legged robots. This is done via the use of a “stratified” configuration space concept and “stratified control theory.” Hence, quasi-static legged locomotors can be considered as extended nonholonomic systems. The paper’s second goal is to explore this theory in the context of “minimalist” legged robots—ones whose legs incorporate fewer than the typical number of degrees-of-freedom (DOF).

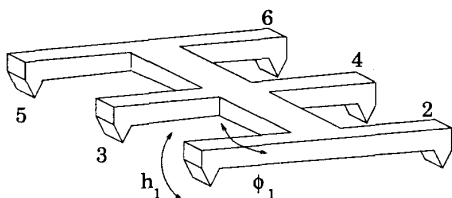


Figure 1: Schematic of minimalist hexapod robot.

The hexapod shown in Fig. 1 and explored in Section 6 is an example of minimalist legged robot. Each leg has only two DOF—the robot can only lift its legs up and down and move them forward and backward. Hexapods are conventionally designed with three independent DOF per leg. This design’s limited control authority may be practically desirable because it decreases the robot’s mechanical complexity. This leg geometry can also probably be implemented at very small

size scales using MEMS technology. However, such decreased kinematic complexity comes at the cost of requiring more sophisticated control and motion planning theory. Note that it is not immediately clear if this robot can move “sideways.” The issue of this mechanism’s ability to move sideways is the *controllability problem*. We review the controllability theory for such systems that we developed in Ref.s [1, 2]. We then consider how to plan the robot’s joint movements so that it can follow a given trajectory. A conventional “foot-placement” approach, where the foot must be placed as necessary to implement vehicle motion, will clearly not work for the robot in Fig. 1, as sideways leg placement is impossible. This paper summarizes a “stratified” motion planning theory which handles both minimalist and conventional quasi-static legged locomotors in a unified way. While we do not explore this issue here, the method can also be applied to many kinematic models of multi-fingered manipulation.

Our motion planning method is independent of the number of legs and many other aspects of a robot’s morphology, and is not based on foot placement concepts. Our approach, which focuses on control inputs, is motivated by the trajectory generation method of Lafferriere and Sussmann [3] for a class of smooth nonlinear kinematic systems. However, the equations of motion of legged robots are not smooth since they cyclically make and break contact. Hence, the method of Ref. [3] can not be directly applied. We extend the approach of Ref. [3] by using a stratified c-space structure in a novel way. It is likely that other methods for steering smooth systems (such as Ref. [4]) can be similarly extended via our framework. *A main contribution of this work is the geometric framework that supports the extension of nonholonomic motion planning techniques to these systems.*

For a given quasi-static legged robot, one might develop a specific motion planner that would perform as well, or possibly better, than the our technique. The key advantage of this approach is its generality. It is particularly useful for quickly roughing out a planner during the preliminary stages of legged robot system

design. More importantly, for minimalist systems, it is the only general planner of which we are aware.

There is a vast literature on legged robotic locomotion. Prior efforts have often focused on a particular morphology (e.g. biped [5], quadruped [6, 7], or hexaped [8]) or on a particular locomotion modality (e.g. quasi-static [8] or hopping [9]). Less effort has been devoted to uncovering principles that span all morphologies and assumptions. This work uses the geometry of stratified c-spaces to formulate results which span morphologies. Our work also makes a novel connection with nonlinear geometric control theory. We believe that this connection is a useful step towards establishing a solid basis for locomotion engineering.

2 Background

We assume that the reader is familiar with the basic ideas of differential geometry and nonlinear control theory, as in Ref. [10]. Recall that the equations of motion for smooth kinematic nonholonomic systems take the form of a driftless nonlinear affine control system evolving on a configuration manifold, M :

$$\dot{x} = g_1(x)u_1 + \dots + g_m(x)u_m \quad x \in M. \quad (1)$$

Since we restrict our analysis to quasi-static locomotion, the governing equations of motion will piecewise take the form of Eq. (1) on each strata. The Lie bracket between two vector fields, $g_1(x)$ and $g_2(x)$, is

$$[g_1(x), g_2(x)] = \frac{\partial g_2(x)}{\partial x} g_1(x) - \frac{\partial g_1(x)}{\partial x} g_2(x)$$

and can be interpreted as the leading order term that results from the sequence of flows

$$\phi_\epsilon^{-g_2} \circ \phi_\epsilon^{-g_1} \circ \phi_\epsilon^{g_2} \circ \phi_\epsilon^{g_1}(x) = \epsilon^2 [g_1, g_2](x) + \mathcal{O}(\epsilon^3),$$

where $\phi_\epsilon^{g_1}(x_0)$ represents the solution of the differential equation $\dot{x} = g_1(x)$ at time ϵ starting from x_0 .

The flow along the vector field g_i can be considered by its *formal exponential* of g_i , denoted by

$$\phi_t^{g_i}(x) := e^{tg_i}(x) = (I + tg_i + \frac{t^2}{2}g_i^2 + \dots) \quad (2)$$

where terms of the form g_i^k are partial differential operators. In Eq. (2), composition occurs from left to right, e.g. $\phi_{t_2}^{g_2} \circ \phi_{t_1}^{g_1} = e^{g_1 t_1} e^{g_2 t_2}$. Both sides of this equation mean “flow along g_1 for time t_1 , then flow along g_2 for time t_2 .” The relationship between the flow along vector fields sequentially is given by the Campbell–Baker–Hausdorff formula [11].

Theorem 1. *Given two smooth vector fields g_1, g_2 the composition of their exponentials is given by*

$$e^{g_1} e^{g_2} = e^{g_1 + g_2 + \frac{1}{2}[g_1, g_2] + \frac{1}{12}([g_1, [g_1, g_2]] - [g_2, [g_1, g_2]]) + \dots} \quad (3)$$

2.1 Trajectory generation for smooth systems

Here we review the motion planning method of Ref. [3] for smooth systems which are governed by an equation of the form Eq. (1). A nonholonomic system often does not have enough controls to directly drive each state variable along a given trajectory, i.e., the number m in Eq. (1) is less than the c-space dimension. In the approach of Ref. [3], this deficit is managed by introducing an *extended system*, where *fictitious controls*, corresponding to higher order Lie bracket motions, are added to the original system of Eq. (1),

$$\dot{x} = b_1 v^1 + \dots + b_m v^m + b_{m+1} v^{m+1} + \dots + b_s v^s \quad (4)$$

where $b_i = g_i$ for $i = 1, \dots, m$, and the b_{m+1}, \dots, b_s correspond to higher order Lie brackets of the g_i , chosen so that $\dim(\text{span}\{b_1, \dots, b_s\}) = \dim(T_x M)$. The v^i 's are called *fictitious inputs* since they may not correspond with any actual system inputs. The higher order Lie brackets must belong to the Philip Hall basis for the Lie algebra. The control inputs v^i which steer the extended system can be found as follows. To go from a point p to a point q , define a curve, $\gamma(t)$ connecting p and q . Using $\gamma(t)$, solve

$$\dot{\gamma}(t) = g_1(\gamma(t))v^1 + \dots + g_s(\gamma(t))v^s \quad (5)$$

for the fictitious controls v_i . This will involve inverting a square matrix or determining a pseudo-inverse.

To find the actual control inputs, first determine the Philip Hall basis for the Lie algebra generated by g_1, \dots, g_m , and denote it by B_1, B_2, \dots, B_s . All flows of Eq. (1) can be represented in the form

$$S(t) = e^{h_s(t)B_s} e^{h_{s-1}(t)B_{s-1}} \dots e^{h_2(t)B_2} e^{h_1(t)B_1} \quad (6)$$

for some functions h_1, h_2, \dots, h_s , called the (backward) Philip Hall coordinates. Furthermore, $S(t)$ satisfies the formal differential equation

$$\dot{S}(t) = S(t)(B_1 v_1 + \dots + B_s v_s); \quad S(0) = 1. \quad (7)$$

If we define the *adjoint mapping*

$$\text{Ad}_{e^{-h_i B_i} B_j} = e^{-h_i B_i} B_j e^{h_i B_i},$$

then it is straight-forward to show that [12]

$$\text{Ad}_{e^{-h_i B_i} \dots e^{-h_{j-1} B_{j-1}} B_j} \dot{h}_j = \left(\sum_{k=1}^s p_{j,k}(h) B_k \right) \dot{h}_j, \quad (8)$$

for some polynomials $p_{j,k}(h)$. Equating coefficients of Eq. (7) with the derivative of Eq. (6), and using Eq. (8), yields differential equations having the form

$$\dot{h} = A(h)v \quad h(0) = 0. \quad (9)$$

These equations specify the evolution of the backward Philip Hall coordinates in response to the fictitious inputs, which were found via Eq. (5).

Next one determines the real inputs using the Philip Hall coordinates. It is easier to determine the inputs using the forward rather than backward Philip Hall coordinates. The backward to forward coordinate transformation is an algebraic operation [3]. For systems which are nilpotent of order two, or which are approximated as nilpotent of order two, the forward to backward transform can be avoided. In these cases, the controls can be obtained from the fictitious controls by use of Lie-bracket-like motions as needed. This will often be the case in practice, as physical systems that require Lie bracket motions of order greater than two are difficult to control. For this reason, and for clarity of presentation, we limit our attention to second order brackets. However, there is no theoretical limitation on the order of brackets.

For nilpotent systems, this method exactly steers the system to the desired final state. Else, the system is steered to a point that is, at worst, half the distance to the desired state [3]. The algorithm can be iterated to arbitrary precision. This method also includes the notion of a “critical” step length which is estimated in Ref. [3], and is often best found by simulation.

3 Stratified Configuration Spaces

The method reviewed above can not be used for legged robots because their equations of motion are not smooth. To adapt this and similar nonholonomic motion planning methods to these systems, we use the notion of a *stratified* c-space. Let S_0 denote a robot’s c-space, which describes the robot’s position, orientation, and joint variables. The robot’s possible configurations will be subjected to constraints if one or more feet are in ground contact. The set of configurations corresponding to the i^{th} foot contact is generically a codimension one submanifold of S_0 , denoted by $S_i \subset S_0$. We assume that S_i is locally defined by the level set of a function $\Phi_i(x) : S_0 \rightarrow \mathbb{R}$.

When both the i^{th} and j^{th} feet are on the ground, the corresponding set of states is a codimension 2 submanifold of S_0 that is formed by the intersection of the two single contact submanifolds: $S_{ij} = S_i \cap S_j$. The c-space structure for a biped is abstractly illustrated in Fig. 2. For greater numbers of legs, further intersections, corresponding to more complex

contact states, can be defined in a recursive fashion: $S_{ijk} = S_i \cap S_j \cap S_k = S_i \cap S_{jk}$, etc. We denote an arbitrary intersection set (or “stratum”) by $S_I = S_{i_1 i_2 \dots i_n}$, $I = \{i_1 i_2 \dots i_n\}$, whose codimension is n , the length of the multi-index subscript. We assume that S_I is a regular submanifold of S_0 . This is generically true for rigid body mechanisms, and implies that the functions $\Phi_{i_1}, \Phi_{i_2}, \dots, \Phi_{i_k}$ are independent. This functional independence will be satisfied if the functions Φ_I correspond to foot heights.

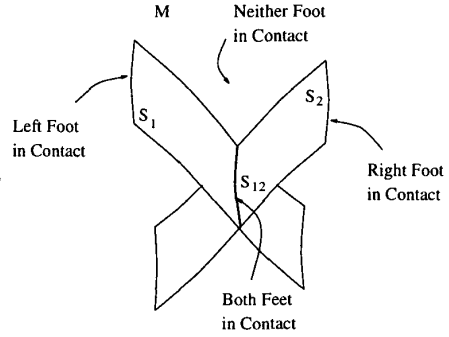


Figure 2: Abstract depiction of the stratified structure of a biped robot c-space.

We say that the robot c-space is *stratified*. Classically, a *regularly stratified* set \mathcal{X} is a set $\mathcal{X} \subset \mathbb{R}^m$ decomposed into a finite union of disjoint smooth manifolds, called *strata*, satisfying the Whitney condition. The strata dimension varies between zero (points) and m (open subsets of \mathbb{R}^m). The Whitney condition requires that the tangent spaces of two neighboring strata “meet nicely.” This condition is generically satisfied for our case (see Ref. [13] for details). Departing slightly from the classical stratification definition [13], we will refer to submanifolds S_i, S_{ij}, S_{ijk} , etc, as strata. We term the highest codimension stratum containing the point x as the *bottom stratum*, and any other submanifolds containing x as *higher strata*. When making comparisons among strata, we will refer to higher codimension strata as *lower strata*, and lower codimension strata as *higher strata*.

The equations of motion at $x \in S_I$ are written as

$$\dot{x} = g_{I,1}(x)u^{I,1} + \dots + g_{I,n_I}(x)u^{I,n_I}, \quad (10)$$

where n_I depends upon the codimension of S_I . When an additional foot contacts the ground, the robot is subjected to additional constraints, and its equations of motion will change in a non-smooth manner. Otherwise, the equations of motion are smooth, though generally different in each strata. Hence, the discontinuities are localized to regions of transition between

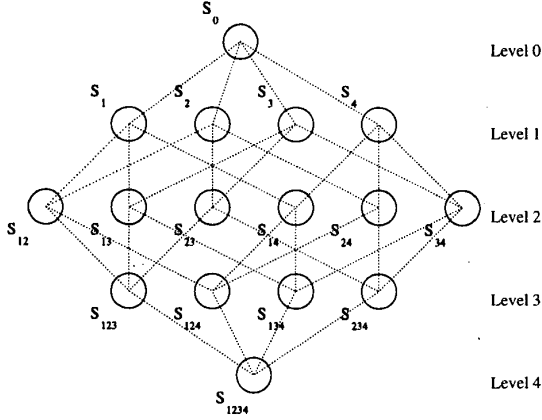


Figure 3: Four Level Stratification

strata. We assume that the control vector fields for any given stratum are well defined at all points in that stratum, including points contained in any substrata of that stratum. They do *not* however represent the system's equations of motion in the substrata, but, nonetheless, are still well defined as vector fields.

Fig. 3 illustrates, via a graph-like structure, a stratification corresponding to a four-legged walker. A node corresponds to a stratum, and the presence of an edge indicates that it is possible to move between the edge-connected strata. The ability to move between two strata depends upon the mechanics of a given problem, and will generally be obvious in a given situation. See Ref. [2] for more details.

We specify a *gait* as an ordered sequence of strata:

$$\mathcal{G} = \{S_{I_1}, S_{I_2}, \dots, S_{I_n}, S_{I_{n+1}} = S_{I_1}\}. \quad (11)$$

The first and last element of the sequence are identical, indicating that the gait is a closed loop in the strata graph. We further assume that the specified gait satisfies the controllability conditions of Ref. [2] so that arbitrary trajectories can be tracked.

Summarizing, we assume that the only discontinuities present in the equations of motion are due to transitions on and off of the strata. We similarly assume that the control vector fields restricted to any stratum are smooth away from points contained in intersections with other strata. When a configuration manifold is consistent with the above description, we will refer to it as a *stratified configuration space*.

Definition 1. Let S_0 be a manifold, and n functions $\Phi_i : S_0 \mapsto \mathbb{R}$, $i = 1, \dots, n$ be such that the level sets $S_i = \Phi_i^{-1}(0) \subset S_0$ are regular submanifolds of S_0 , for each i , and the intersection of any number of level sets,

$S_{i_1 i_2 \dots i_m} = \Phi_{i_1}^{-1}(0) \cap \Phi_{i_2}^{-1}(0) \cap \dots \cap \Phi_{i_m}^{-1}(0)$, $m \leq n$, is also a regular submanifold of S_0 . Then S_0 and the functions Φ_n define a *stratified c-space*.

4 Controllability

For a given strata, S_I , the *distribution* defined by the span of the control vector fields active on S_I is:

$$\Delta_{S_I} = \text{span} \{g_{S_{I,1}}, \dots, g_{S_{I,n_I}}\}.$$

The *involutive closure* of Δ_{S_I} , denoted by $\overline{\Delta}_{S_I}$, is the closure of Δ_{S_I} under Lie bracketing. The controllability of a given gait, Eq. (11), can be determined as follows. Let $\mathcal{D}_1 = \overline{\Delta}_{I_1}$. If $S_{I_{i-1}} \subset S_{I_i}$, then $\mathcal{D}_i = \mathcal{D}_{i-1} + \overline{\Delta}_{I_i}$. Else, if $S_{I_i} \subset S_{I_{i-1}}$, then $\mathcal{D}_i = (\mathcal{D}_{i-1} \cap TS_{I_i}) + \overline{\Delta}_{I_i}$. In Ref. [2] it is shown that if $\dim(\mathcal{D}_n) = \dim(T_{x_0} S_B)$ the system is *gait controllable* from x_0 (i.e., the system can reach an open nbhd of x_0 in the bottom strata). This is the extension of the classical Chow's theorem for the small time local controllability of nonholonomic systems to the stratified case. For a more rigorous discussion of stratified system controllability, see Ref.s [1, 2].

5 Legged Trajectory Generation

This section extends the procedure of Section 2 to kinematic systems having a stratified c-space. Assume that the robot starts at a configuration p and seeks to reach configuration q . We assume that both p and q lie in the same bottom stratum, denoted by S_B . This corresponds to the robot starting and stopping with the same set of feet in ground contact. Eliminating this requirement is a simple extension of this work. We extend the method of Section 2.1 to legged and fingered robotic systems via the notion of a *stratified extended system* on S_B .

5.1 The Stratified Extended System

Usually, the goal q can not be reached by remaining in S_B —some switching amongst the strata will be necessary. While the equations of motion in S_B will generally be different than those in higher strata, the equations of motion in the higher strata are valid at points arbitrarily close to S_B since S_B is defined by the intersection of higher strata. As shown in the example below, it is possible to consider the vector fields associated with each stratum in one common space (additional examples that deal with more subtle issues can be found in Ref. [14]). In this case, the common space is S_B . This concept leads to the definition of a “stratified extended system.”

Example 1. Consider the conceptual biped c-space shown in Fig. 2. Assume that on stratum S_{12} , the

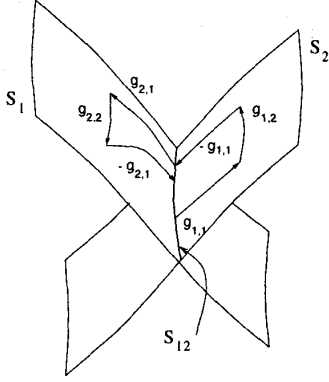


Figure 4: Sequence of Flows

vector field $g_{1,1}$ moves the system off of S_{12} and onto S_1 , and correspondingly, $g_{2,1}$ moves the system off of S_{12} onto S_2 . Also, we consider the vector fields $g_{1,2}$ and $g_{2,2}$, defined on S_1 and S_2 respectively. Consider the following sequence of flows, starting from $x_0 \in S_{12}$

$$x_f = \underbrace{\phi_{-g_{2,1}}^{t_6}}_{S_{12} \leftarrow S_2} \circ \underbrace{\phi_{g_{2,2}}^{t_5}}_{\text{on } S_2} \circ \underbrace{\phi_{g_{2,1}}^{t_4}}_{S_2 \leftarrow S_{12}} \circ \underbrace{\phi_{-g_{1,1}}^{t_3}}_{S_{12} \leftarrow S_1} \circ \underbrace{\phi_{g_{1,2}}^{t_2}}_{\text{on } S_1} \circ \underbrace{\phi_{g_{1,1}}^{t_1}}_{S_1 \leftarrow S_{12}}(x_0). \quad (12)$$

See Fig. 4. The notation " $S_{12} \leftarrow S_1$ " means that the flow takes the system from S_1 to S_{12} and "on S_1 " means that the flow lies entirely in S_1 .

From Eq. (3), if the Lie bracket between two vector fields is zero, their flows commute. Thus, if

$$[g_{1,1}, g_{1,2}] = 0 \quad \text{and} \quad [g_{2,1}, g_{2,2}] = 0, \quad (13)$$

we can reorder the above sequence of flows, by interchanging the flow along $g_{1,1}$ and $g_{1,2}$ and the flows along $g_{2,1}$ and $g_{2,2}$ as follows

$$x_f = \underbrace{\phi_{g_{2,2}}^{t_5} \circ \phi_{-g_{2,1}}^{t_6}}_{\text{interchanged}} \circ \underbrace{\phi_{g_{2,1}}^{t_4} \circ \phi_{-g_{1,1}}^{t_3}}_{\text{interchanged}} \circ \underbrace{\phi_{g_{1,2}}^{t_2} \circ \phi_{g_{1,1}}^{t_1}}_{\text{on } S_{12}}(x_0), \quad (14)$$

if $t_1 = t_3$ and $t_4 = t_6$. Note that $g_{1,2}$ and $g_{2,2}$ are vector fields in the equations of motion for strata S_1 and S_2 , respectively, but not on stratum S_{12} . However, the sequence of flows in (12) occurs on different strata, where the flows are governed by vector fields associated with each stratum. This flow yields the same net result as the net flow in Eq. (14), where the vector fields are evaluated on the bottom stratum, even though they are

not part of the equations of motion there. Furthermore, we note that if $g_{1,2}$ and $g_{2,2}$ are tangent to S_{12} , the resulting flow in Eq. (14) will remain in S_{12} . In fact, it is implicitly required in the above argument that at least $g_{1,2}$ is tangent to S_{12} .

If the bottom stratum is described by the level set of a function, Φ_B , and if $g_{1,2}$ is not tangent to S_B , then, $\langle d\Phi_B, g_{1,2} \rangle = f_1(x) \neq 0$. Also, since the vector field $g_{1,1}$ moves the foot out of contact, we similarly have $\langle d\Phi_B, g_{1,1} \rangle = f_2(x) \neq 0$. Then, the vector field, $\tilde{g}_{1,2} = g_{1,2} - \frac{f_1(x)}{f_2(x)} g_{1,1}$, is tangent to S_B because

$$\langle d\Phi_B, \tilde{g}_{1,2} \rangle = \langle d\Phi_B, g_{1,2} \rangle - \frac{f_1(x)}{f_2(x)} \langle d\Phi_B, g_{1,1} \rangle = 0.$$

Henceforth, we assume that a vector field on the higher stratum is tangent to the lower stratum, otherwise we can modify it to be so in the above manner.

In Example 1 and similar examples in [14, 15], certain Lie brackets must be zero. While one could check that these conditions are met in a given problem, the following assumption guarantees this condition.

Assumption 2. If a foot must be lifted from the ground during a gait, we assume that the robot can control, (via a single control, or a combination of controls), the height of that foot relative to the ground. We further assume that for each stratum comprising a gait, the system's equations of motion are independent of foot height. In this case, the Lie bracket of the vector field controlling foot height with any other vector field is zero, so that the decoupling requirement is satisfied. Additionally, the tangency requirements for cancelling the flows associated with raising and lowering the foot will automatically be satisfied.

This is arguably a strict assumption. However, for kinematic, legged robots this assumption will almost always be satisfied (see Section 6 for an example).

The example above and others in [14, 15] show that vector fields in higher strata can be effectively considered as part of the equations of motion for the system on the bottom stratum. Based on this observation, we introduce the following.

Definition 3. The *extended stratified system* on S_B is the driftless affine system comprised of the vector fields on S_B , chosen vector fields from the higher strata, and Lie brackets of vector fields from S_B and higher strata. I.e., it is a system taking the form:

$$\begin{aligned} \dot{x} = & b_1(x)v_1 + \cdots + b_m(x)v_m + \underbrace{b_{m+1}v_{m+1} + \cdots + b_nv_n}_{\text{from higher strata}} \\ & + \underbrace{b_{n+1}v_{n+1} + \cdots + b_pv_p}_{\text{any Lie brackets}}, \end{aligned} \quad (15)$$

where the $\{b_1, \dots, b_p\}$ span $T_x S_0$.

With this definition, we have effectively increased the class of vector fields that we may employ when using the algorithm presented in Section 2.

5.2 The Motion Planning Algorithm

Our trajectory planning method is based on the use of the method of Section 2.1 with the stratified extended system. The stratified extended system is used to plan motions in the bottom stratum, thereby obtaining fictitious inputs. The actual inputs are found by the method in Section 2.1, with the modification that when the system must flow along a vector field in a higher stratum, it switches to that stratum by lifting the appropriate feet, flowing along the vector field, and then replacing the appropriate feet, as in Example 1. Specifically, the algorithm to generate trajectories that move the robot from p to q follows.

1. Construct the *extended stratified system*, Eq. (15), on the bottom strata, S_B .
2. Find a nominal trajectory, $\gamma(t)$, that connects p and q . Given $\gamma(t)$, solve

$$\dot{\gamma}(t) = b_1(x)v_1 + \dots + b_p(x)v_p,$$

for the fictitious inputs, v_i . It may be desirable (or necessary for gait stability [14, 15]) to decompose the trajectory into subtrajectories.

3. Solve the stratified extended system for the fictitious control inputs.
4. For each path segment in each strata, compute the real inputs that steer the robot along $\gamma(t)$.
5. Flow along each first order vector field, and approximate higher order vector fields as illustrated in Example 1. In general, it will be necessary to switch strata between some of these flows.

The straight-forward application of this adaption of Section 2.1 may result in an inordinate number of strata switches because the sequence of flows in Eq. (6) are arranged by *order*. From a gait efficiency point of view, it is desirable to have them arranged by strata. It is possible to regroup the flow sequence by strata if the Lie bracket between vector fields from different strata are zero. Flows corresponding to the same stratum can be grouped together, thereby reducing the number of contact transitions.

There is not an inherent mechanism in the application of the method of Section 2.1 to guarantee gait

stability. Recall that the fictitious inputs are determined from an extended system trajectory, $\gamma(t)$. The realized trajectory will generally *not* be $\gamma(t)$. Thus, it is not sufficient to pick an initial trajectory $\gamma(t)$ which is always stable. One also must guarantee that the method's inherent deviations from the initial trajectory lie within the stability bounds. This issue is discussed in more detail in [14, 15], where it is shown that modifications of the algorithm's step length can insure stability, as well as obstacle avoidance.

6 Example

We illustrate our approach by steering the hexapod of Fig. 1 to walk over flat terrain. The key difficulty is the fact that the legs are kinematically insufficient, making sideways motion difficult. Assume that the robot walks with a tripod gait¹, alternating movements of legs 1-4-5 with movements of legs 2-3-6. With the tripod gait, this robot has four control inputs: inputs u_1 and u_2 respectively control the forward/backward angular leg displacements of legs 1-4-5 and legs 2-3-6, while inputs u_3 and u_4 respectively control the height of legs 1-4-5 and 2-3-6.

The equations of motion can be written as follows.

$$\begin{aligned} \dot{x} &= \cos \theta (\alpha(h_1)u_1 + \beta(h_2)u_2) \\ \dot{y} &= \sin \theta (\alpha(h_1)u_1 + \beta(h_2)u_2) \\ \dot{\theta} &= l\alpha(h_1)u_1 - l\beta(h_2)u_2 \\ \phi_1 &= u_1; \quad \phi_2 = u_2 \\ \dot{h}_1 &= u_3 \quad \dot{h}_2 = u_4 \end{aligned}$$

where (x, y, θ) represents the body's configuration, ϕ_1 (ϕ_2) is the angle of legs 1-4-5 (legs 2-3-6), l is the leg length, and h_i is the leg's height off the ground. The functions $\alpha(h_1)$ and $\beta(h_2)$ are defined by

$$\alpha(h_1) = \begin{cases} 1 & \text{if } h_1 = 0 \\ 0 & \text{if } h_1 > 0 \end{cases} \quad \beta(h_2) = \begin{cases} 1 & \text{if } h_2 = 0 \\ 0 & \text{if } h_2 > 0 \end{cases}$$

Stability is ensured in the tripod gait if the robot's center of mass remains above the triangle defined by the feet which are in ground contact. For the motion of legs 1-4-5, the robot's center of mass must be at least $b = \frac{l_b}{4} + l \sin \phi_1$ from the robot's front to ensure stability, where l_b denotes the body's length. See Fig. 5. Alternatively, if the center of mass is located a distance b from the robot's front, stability is ensured during the motion if both of these constraints are satisfied

$$\phi_1 < \sin^{-1}\left(\frac{b}{l} - \frac{l_b}{4l}\right) \quad \phi_2 > -\sin^{-1}\left(\frac{3l_b}{4l} - \frac{b}{l}\right).$$

¹Ref. [1] shows that the hexapod is small time locally gait controllable when a tripod gait is used.

which yields.

$$\begin{aligned} h_1(1) &= 1/2 & h_2(1) &= 1/2 \\ h_3(1) &= -1/2 & h_4(1) &= -1/2 & h_5(1) &= 3/4. \end{aligned}$$

Since the nilpotent approximation is of order two, there is no need to transform to forward Philip Hall coordinates. Let \circ denote concatenation of control inputs. For example, $u_1 \circ u_2$ denotes that $u_1 = 1$ for time $h_1(1)$ followed by $u_2 = 1$ for time $h_2(1)$. The system must flow along $g_{12,1}, g_{12,2}$ for $\frac{1}{2}$ seconds, and we must construct a piece-wise approximation to the flow along $[g_{12,1}, g_{12,2}]$ for $\frac{3}{4}$ seconds. The sequence that approximately moves the system in this bracket direction is

$$u_1 \circ u_2 \circ -u_1 \circ -u_2 \quad (20)$$

where each control input equals one for $\sqrt{\frac{3}{4}}$ seconds. To flow along $g_{11,1}$, $u_1 = 1$ for $\frac{1}{2}$ seconds. Similarly to flow along $g_{12,1}$, $u_2 = 1$ for $\frac{1}{2}$ seconds.

On the higher strata, to flow along $g_{1,1}$, $u_1 = -1$ for $\frac{1}{2}$ seconds and to flow along $g_{2,1}$, $u_1 = -1$ for $\frac{1}{2}$ seconds. To execute these flows, the robot must switch from S_B to a higher strata when executing a control input associated with a fictitious input for the higher strata. Thus, the total control sequence is

$$\begin{aligned} &\sqrt{\frac{3}{4}}(u_1 \circ u_2 \circ -u_1 \circ -u_2) \\ &\quad \circ \frac{1}{2}u_2 \circ \frac{1}{2}u_1 \circ \epsilon u_4 \circ (-\frac{1}{2}u_2) \circ (-\epsilon u_4) \\ &\quad \circ \epsilon u_3 \circ (-\frac{1}{2}u_1) \circ (-\epsilon u_3). \end{aligned}$$

The first four terms approximate the Lie bracket motion on S_{12} . The $\sqrt{\frac{3}{4}}$ term denotes the length of time each control input is “on.” The next two terms are the contribution of the u_1 and u_2 terms individually on S_{12} . The next term represents a small flow associated with legs 2-3-6 breaking ground contact, and the following term corresponds to legs 2-3-6 moving back to their initial position. Since the legs are not in ground contact, this motion does not cause the robot’s body to move. The next input corresponds to legs 2-3-6 making ground contact. The next three inputs correspond to legs 1-4-5 performing an analogous motion.

Fig. 6 shows the path of the robot’s center as it follows a straight line trajectory, which is broken into four equal segments. Due to the nilpotent approximation, there is some final error. Better accuracy can be obtained by use of a higher order nilpotent approximation or a second algorithm iteration from the robot’s ending position. Note that the body is oriented along the x -axis in this example. Since the legs can not move

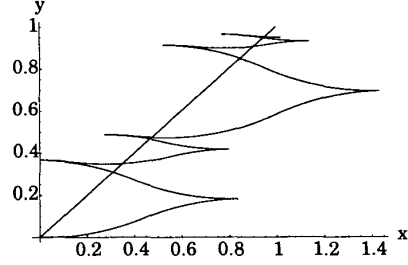


Figure 6: Straight Trajectory

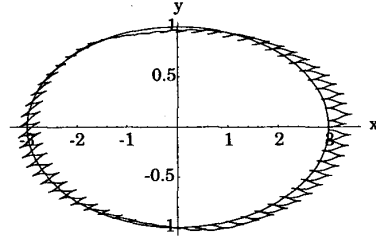


Figure 7: Elliptical Trajectory with Smaller Steps

immediately sideways, the robot’s motion must include “parallel-parking-like” behavior.

There is no inherent need to break the trajectory into subsegments, however, there are reasons to do so. First, since the method is based upon decomposing a desired trajectory into flows along the Philip Hall basis vector fields, the final trajectory is only related to the desired trajectory in that the end points are the same (or approximately the same for nilpotent approximations). Breaking the path into segments leads to better overall tracking. Second, robot stability requirements may also demand smaller steps.

The approach is general enough to enable approximate tracking of arbitrary trajectories. Fig. 7 shows the hexapod following an ellipse while maintaining a constant angular orientation. In the simulation, the elliptical trajectory is broken into 60 segments. In this example, part of the trajectory tracking error is due to the nilpotent approximation, but another contribution to the error is the simplicity of the model. Some directions are more “difficult” for the system to execute than others due to the kinematic limitations of the leg design. It can be shown that the stability criteria is satisfied along this path.

Fig. 8 depicts the footprints left by the hexapod as it follows a straight line diagonal path while simultaneously rotating at a constant rate. The complex footfall pattern suggests that techniques based on foot placement would be difficult to apply to this system.

Finally, we consider an obstacle avoidance example (Fig. 9). A set of walls are indicated by dark grey

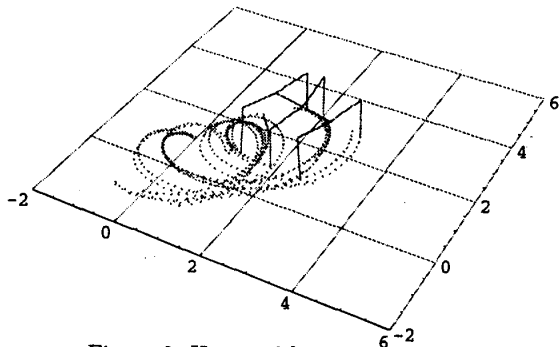


Figure 8: Hexapod leaving footprints.

regions. The lighter grey regions correspond to locations of the vehicle's center where some vehicle orientations may cause the hexapod to intersect the walls. To make the problem more challenging, the robot must uniformly rotate as it follows the nominal trajectory. While the nominal initial trajectory $\gamma(t)$ must *a priori* avoid any obstacles, this constraint alone will not guarantee that the actual motion avoids obstacles. Since the path of the center of mass intersects the lighter grey regions during portions of its motion, the robot would realistically bump into the walls in this example. However, a minor further subdivision of the trajectory according to the method outlined in Ref.s [14, 15] results in a wall avoiding trajectory.

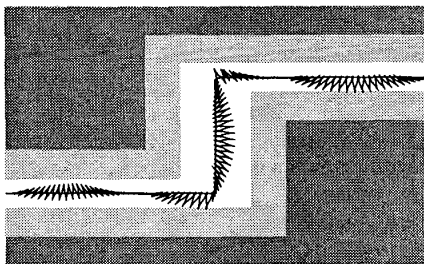


Figure 9: Obstacle avoidance example

7 Conclusions

Using stratified c-space notions, we are able to show that quasi-static legged locomotors can be analyzed and guided by extensions of methods that were originally developed for nonholonomic wheeled vehicles. The general motion planning scheme which arises from this approach works for many types of legged robotic systems independently of the number of legs or leg morphology. The simulations indicate that the approach is rather simple to apply. More importantly, we believe that our approach provides an evolutionary path for future research and generalizations.

Acknowledgments: This work was partially supported by the National Science Foundation through an Engineering Research Center grant, and a grant from the Office of Naval Research.

References

- [1] Bill Goodwine and Joel Burdick. Gait controllability for legged robots. In *Proc IEEE Int. Conf on Robotics and Automation*. IEEE, 1998.
- [2] B. Goodwine and J.W. Burdick. Controllability of kinematic systems on stratified configuration spaces. *IEEE Trans. Automatic Control* (to appear), 2000.
- [3] G. Lafferriere and Hector J. Sussmann. A differential geometric approach to motion planning. In X. Li and J. F. Canny, editors, *Nonholonomic Motion Planning*, pages 235–270. Kluwer, 1993.
- [4] R.M. Murray and S.S. Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Trans. Automatic Control*, 38:700–716, 1993.
- [5] S. Kajita and K. Tani. Study of dynamic biped locomotion on rugged terrain. In *IEEE Int. Conf. on Robotics and Automation*, pages 1405–1411, Sacramento, CA, 1991.
- [6] J.K. Lee and S.M. Song. Path planning and gait of walking machine in an obstacle-strewn environment. *J. Robotics Systems*, 8:801–827, 1991.
- [7] Matt Berkemeier. Modeling the dynamics of quadrupedal running. *Int. J. Robotics Research*, 16(9):971–985, 1998.
- [8] S.M. Song and K.J. Waldron. *Machines that walk: the Adaptive Suspension Vehicle*. MIT Press, 1989.
- [9] M.H. Raibert. *Legged Robots that Balance*. MIT Press, 1986.
- [10] Alberto Isidori. *Nonlinear Control Systems*. Springer-Verlag, second edition, 1989.
- [11] V.S. Varadarajan. *Lie Groups, Lie Algebras, and Their Representations*. Springer-Verlag, 1984.
- [12] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., 1994.
- [13] Goresky and Macpherson. *Stratified Morse Theory*. Springer-Verlag, New York, 1980.
- [14] J. William Goodwine. *Control of Stratified Systems with Robotic Applications*. PhD thesis, California Institute of Technology, 1998.
- [15] J.W. Goodwine and J.W. Burdick. Motion planning for kinematic stratified systems with application to quasi-static legged locomotion and finger gaiting. In *Proc. Workshop Algorithmic Foundations of Robotics*, Hanover, NH, Mar. 2000.